Análise do desempenho de diferentes vetorizadores e diferentes classificadores na identificação de espécies de anuros através do som

Natanael Monteiro Pastore Antonioli

Por que identificar sons de anuros?

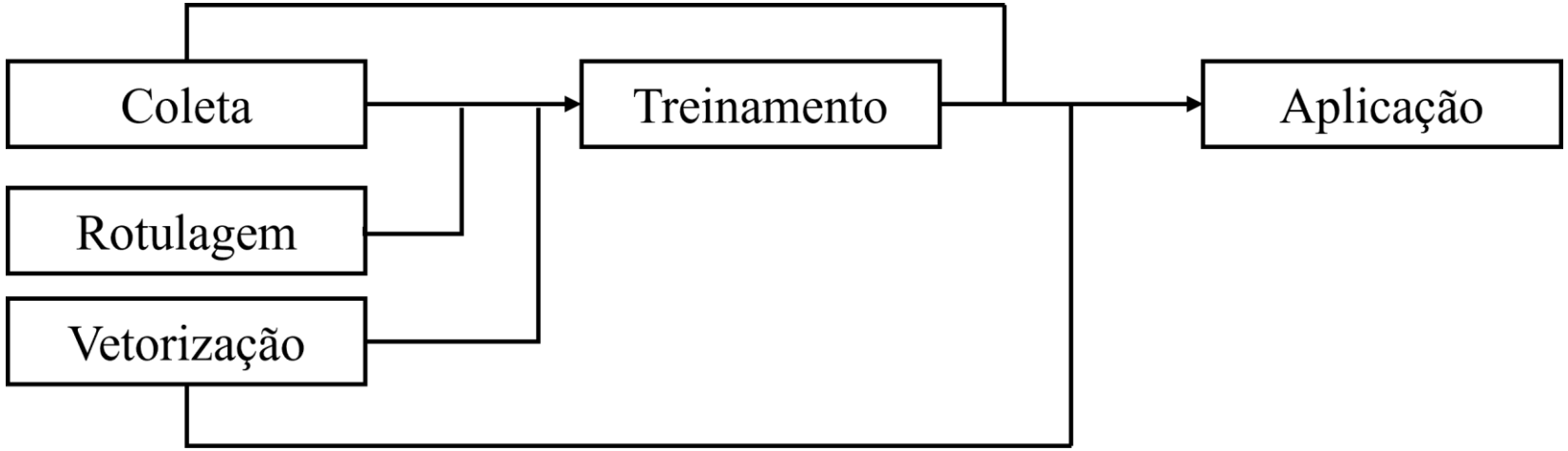
- Anuros incluem todos os anfíbios sem cauda (sapos, rãs e pererecas).
- Pele capaz de absorver água e gases os torna sensíveis a mudanças ambientais.
- Anuros são ótimos marcadores da qualidade de um ecossistema.
- Anuros podem se tornar espécies invasivas.
- Criação de ferramenta de propósito geral para pesquisa ecológica.

Fonte: Getty Images

Objetivo geral e objetivos específicos

- Um sistema com aprendizado de máquina requer uma função de vetorização e uma função de classificação.

- O par de funções é inserido em um sistema de identificação.



Fonte: Elaborado pelo autor com recursos de FlatIcon.

Transformada curta de Fourier (STFT)

- Média da amplitude ao longo do tempo em um conjunto discreto de frequências.

1. Obter a STFT

$$S_1[f,t] = \sum_{n=0}^{M-1} S[n+tH] \times w_{Hann}(n) \times e^{-\frac{i2\pi fn}{M}}$$

$$w_{Hann}(n) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right)$$

2. Obter o valor absoluto.

$$S_2[f,t] = |S_1[f,t]|$$

3. Obter a média ao longo do tempo para cada frequência.

$$v[f] = \frac{1}{T}\sum_{t=0}^{T-1} S_2[f,t]$$

Coeficientes cepstrais de frequência mel (MFCCs)

- Média da potência ao longo do tempo em frequências associadas com a audição humana.

1. Obter a STFT

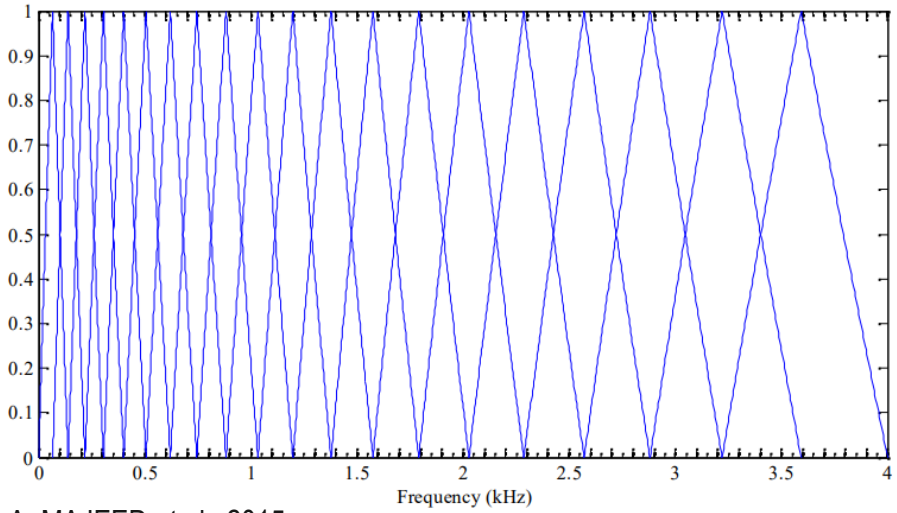
$$S_1[f,t] = \sum_{n=0}^{M-1} S[n+tH] \times w_{Hamming}(n) \times e^{-\frac{i2\pi fn}{M}}$$

$$w_{Hamming}(n) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi n}{N-1}\right)$$

2. Obter a potência

$$S_2[f,t] = \frac{|S_1[f,t]|^2}{N}$$

3. Aplicar os filtros da frequência mel (*melody*).



Fonte: A. MAJEED et al., 2015

$$S_3[f,t] = \log_{10}\left(\sum_{k=0}^{N-1} S_2[f,t] \times \mu[f_{mel},t]\right)$$

Coeficientes cepstrais de frequência mel (MFCCs)

- Média da potência ao longo do tempo em frequências associadas com a audição humana.

4. Realizar a transforma direta cosseno.

$$S_3[f,t] = \sum_{u=1}^{U} S_2 \cos\left(\frac{\pi t}{U}\left(u - \frac{1}{2}\right)\right)$$

6. Obter a média.

$$v[f] = \frac{1}{T}\sum_{t=0}^{T-1} \frac{S_3[f,t] + S_3'[f,t] + S_3''[f,t]}{3}$$

5. Obter a primeira e segunda derivadas.

$$S_3'[f,t] = \frac{\sum_{n-1}^{N} n \times (S_3[f,t+n] - S_3[f,t-n])}{2 \times \sum_{n-1}^{N} n^2}$$

$$S_3''[f,t] = \frac{\sum_{n-1}^{N} n \times (S_3'[f,t+n] - S_3'[f,t-n])}{2 \times \sum_{n-1}^{N} n^2}$$

Rastreio de altura

- Estatísticas importantes sobre a amplitude nas frequências nas quais ela é maior.

1. Obter a STFT

$$S_1[f,t] = \sum_{n=0}^{M-1} S[n+tH] \times w_{Hann}(n) \times e^{-\frac{i2\pi fn}{M}}$$

$$w_{Hann}(n) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right)$$

2. Obter a matriz de gradientes.

$$S_G[f,t] = \frac{S[f,t+1] - S[f,t-1]}{2}$$

Rastreio de altura

➢ Estatísticas importantes sobre a amplitude nas frequências nas quais ela é maior.

3. Obter a matriz de pontos de inflexão através de interpolação parabólica.

$$(x_n, y_n) = (t + n, S[t + n, f])$$

$$(x_{n+1}, y_{n+1}) = (t + n + 1, S[t + n + 1, f])$$

$$(x_{n+2}, y_{n+2}) = (t + n + 2, S[t + n + 1, f])$$

$$\begin{cases} ax_n^2 + bx_n + c = y_n \\ ax_{n+1}^2 + bx_{n+1} + c = y_{n+1} \\ ax_{n+2}^2 + bx_{n+2} + c = y_{n+2} \end{cases} \qquad S_V[f,t] = -\frac{b}{2a}$$

4. Obter as frequências-chave.

$$S_2[f, t] = \frac{1}{2} \times S_G[f, t] \times S_V[f, t]$$

5. Calcular a média no tempo de cada frequência-chave.

$$S_3[f] = \frac{1}{T} \sum_{t=0}^{T-1} S_2[f, t]$$

6. Calcular o vetor final.

$$v = [\overline{S_3}, \sigma_{S_3}, \max S_3\ , \min S_3\ , \widetilde{S_3}, Q_1(S_3), Q_3(S_3)]$$

Croma

- Estatísticas importantes sobre as intensidades em 12 classes de notas.

1. Obter o quadrado do módulo de uma STFT.

$$S_1[f,t] = \left| \sum_{n=0}^{M-1} S[n+tH] \times w_{Hann}(n) \times e^{-\frac{i2\pi fn}{M}} \right|^2$$

$$w_{Hann}(n) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right)$$

2. Aplicar um banco de filtros pré-definido com 12 classes.

$$S_2[f,t] = \sum_{c=0}^{11} C[c,k] \times S_1[f,t\,]$$

3. Normalizar o sinal.

$$S_3[f,t] = \frac{S_2[f,t]}{\sum_{f'=0}^{11} S_2[f',t] + \epsilon}$$

4. Calcular o vetor final.

$$v = [\overline{S_3}, \sigma_{S_3}, \max S_3\,]$$

Energia

- Variação da energia do sinal ao longo do tempo.

1. Obter o vetor de energias usando a janela quadrada.

$$S_1[n] = \sqrt{\frac{1}{L}\sum_{n=0}^{L-1}\left(S[n+mH] \times w_{quadrada}[n]\right)^2}$$

$$w_{quadrada}(n) = \begin{cases} 1, & 0 \leq n < L \\ 0, & \text{caso contrário} \end{cases}$$

2. Realizar o corte do vetor para fixa-lo em R elementos.

Se o vetor tem menos que R elementos, adicionar zeros.
Se o vetor tem exatamente R elementos, deixar como está.
Se o vetor tem mais que R elementos, cortar adicionais.

Bayes ingênuo

- Cada instância é associada a vetores com probabilidades para cada classe, o maior produto dentro do vetor determina o rótulo.

1. Criar k vetores contendo a probabilidade de cada atributo pertencer à cada classe.

$$C_1(c_1, \dots, c_n)$$
$$C_2(c_1, \dots, c_n)$$
$$\dots$$
$$C_k(c_1, \dots, c_n)$$

2. Escolher a classe que maximiza $\hat{y}$.

$$\hat{y} = \max p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

Máquina de vetores de suporte

- Hiperplanos são inseridos em um espaço vetorial e a classificação é decidida por voto majoritário.

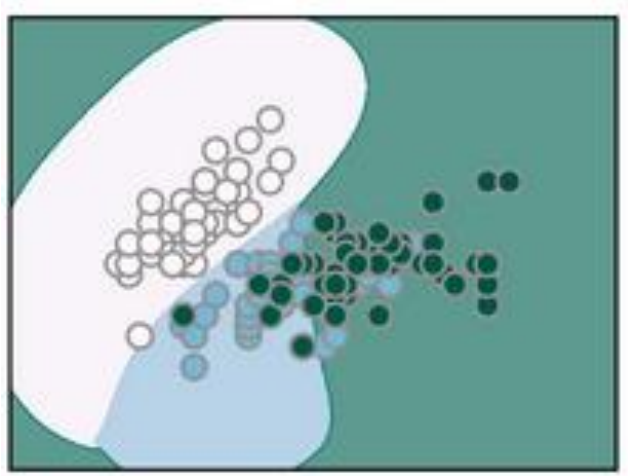
1. Gerar um hiperplano para cada par de classes com uma transformação de núcleo RBF.

$$w^T \cdot x + b = 0$$

$$K(x_1, x_2) = e^{-\gamma \|x_1 - x_2\|^2}$$

2. Cada hiperplano "vota" entre duas classes.

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_1 x_2) + b$$

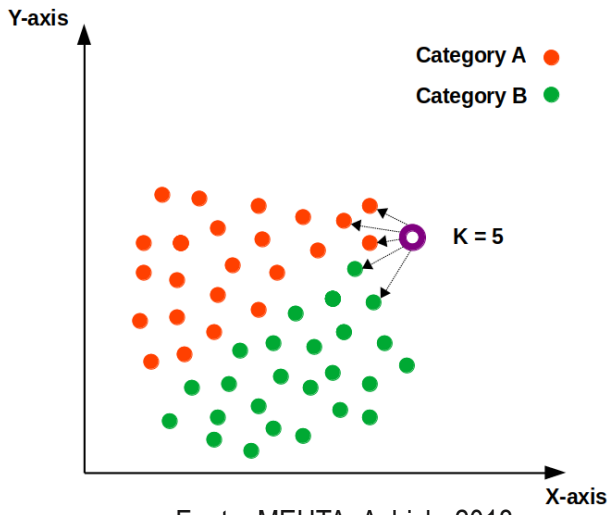


Fonte: MARIUS, Hucker, 2020

k-vizinhos mais próximos (kNN)

- Instâncias são inseridas em um espaço vetorial e a classificação é decidida pela classe dominante na vizinhança.

1. Colocar cada instância de treinamento em um espaço vetorial, assim como a instância a ser classificada.



Fonte: MEHTA, Ashish, 2018

2. Computar a distância para cada instância.

$$d(\mathrm{X}, \mathrm{x}_i) = \sqrt{\sum_{i=0}^{n} (X_i - x_i)^2}$$

3. Computar voto majoritário entre k menores distâncias.

Análise de discriminante quadrático (QDA)

- Classe definida como a maior probabilidade entre a média e a covariância de todas as classes.

1. Para cada classe, calcular vetor de média e vetor de covariância.

$$\mu_k = \frac{1}{n}\sum_{i-1}^{n} \mathrm{x}_i$$

$$V_k = \frac{1}{n-1}\sum_{i=1}^{n} (\mathrm{x}_i - \mu)(\mathrm{x}_i - \mu)^T$$

2. A classe escolhida é a que maximiza:

$$P(\mathrm{X}|y_k) = \frac{1}{2\pi^{1/2}|\mathrm{V}_k|^{1/2}} \times e^{\left(-\frac{1}{2}(\mathrm{X}-\mu_k)^T V^{-1}(X-\mu_k)\right)}$$

Perceptron multicamadas

➢ Classe definida pelo resultado de uma rede neural treinada em sucessivas iterações.

1. Cada *perceptron* tem seu interior e saída configurados.

$$z = \mathrm{w}^T \mathrm{v} + b = b + \sum_{i=0}^{n} w_i v_i$$

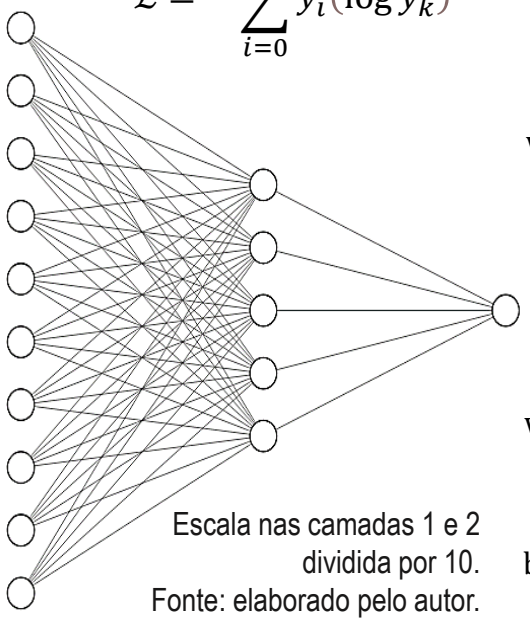
$$\phi(z) = \max(0, z)$$

2. A primeira camada (100), segunda (50) e terceira (1) são configuradas.

$$\mathrm{a}_1 = \max(0, w_1 X + \mathrm{b}_1)$$
$$\mathrm{a}_2 = \max(0, w_2 a_1 + \mathrm{b}_2)$$
$$\mathrm{z} = w_3 a_2 + \mathrm{b}_3$$

3. A rede neural é treinada para minimizar a perda.

$$\mathcal{L} = -\sum_{i=0}^{K} y_i (\log \hat{y}_k)$$



Escala nas camadas 1 e 2 dividida por 10.
Fonte: elaborado pelo autor.

$$\delta_2 = \hat{y} - y$$

$$\delta_1 = (\mathrm{w}_2 \delta_2) \circ (\phi'(z))$$

$$\mathrm{w}_1 \leftarrow \mathrm{w}_1 - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{w}_1} = \mathrm{w}_1 - \eta \delta_1$$

$$\mathrm{b}_1 \leftarrow \mathrm{b}_1 - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{b}_1} = \mathrm{b}_1 - \eta \delta_1 \mathrm{x}$$

$$\mathrm{w}_2 \leftarrow \mathrm{w}_2 - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{w}_2} = \mathrm{w}_2 - \eta \delta_2$$

$$\mathrm{b}_2 \leftarrow \mathrm{b}_2 - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{b}_2} = \mathrm{b}_2 - \eta \delta_2 \mathrm{a}_1$$

4. Do vetor produzido, a maior probabilidade define a classe.

$$\hat{y} = \mathrm{smax}(\mathrm{z})_i = \frac{e^{z_i}}{\sum_{j=0}^{k} e^{z_j}}$$

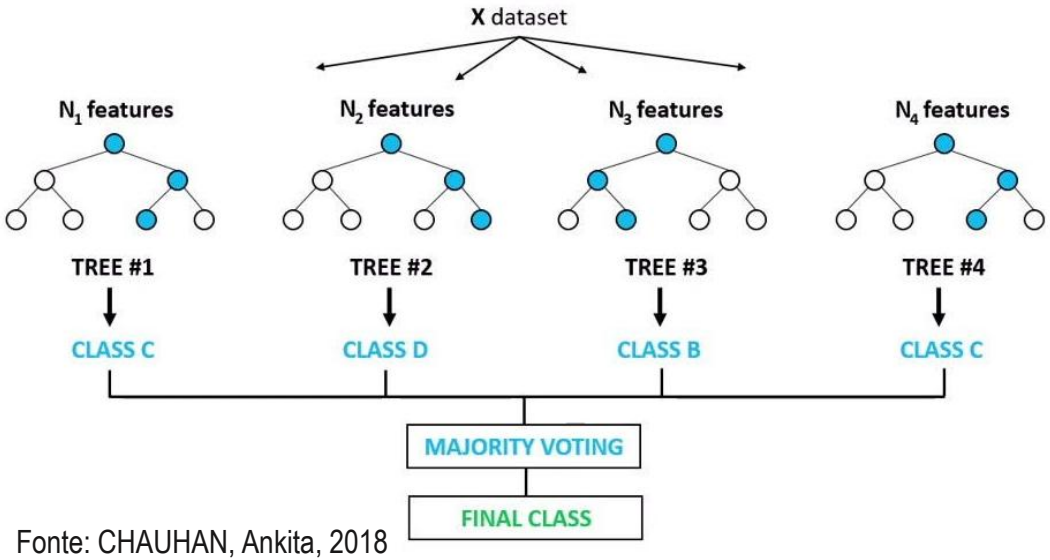
Floresta aleatória

➢ Classe definida pelo voto majoritário entre múltiplas árvores de decisão.

1. Construir certo número de árvores de decisão, cada uma com parte aleatória do conjunto e parte dos atributos.

$$G(t) = 1 - \sum_{i=1}^{K} p_i^2$$

2. Computar o voto majoritário com cada árvore.



Fonte: CHAUHAN, Ankita, 2018

Conjunto de dados

- Contém anuros de todo o mundo.
- Publicação revisada por pares.
- 26 espécies diferentes.
- 1536 segmentos de áudio.
- Aproximadamente 150 segundos para cada espécie.
- Formato *.m4a*.

Anuran Sound (Frogs or Toads) Dataset

Classify anuran sounds into 26 classes

Data Card | Code (0) | Discussion (0) | Suggestions (0)

About Dataset

Abstract

Classification of animal species using animal sounds is a critical issue for bioacoustics work.

Context

The determination of anurans (frogs or toads) species can be used as an indicator of climate change. However, counting and classifying anurans in their natural habitat is challenging. Therefore, computer-assisted intelligent systems must be used to determine anuran types correctly.

Fonte: AKBAL et al., 2023

Ferramentas

Linguagem de programação e plataforma

colab

Vetorização de áudio

librosa

SciPy

NumPy

Organização de dados

pandas

Aprendizado de máquina

Fonte: documentação de cada ferramenta.

Algoritmo

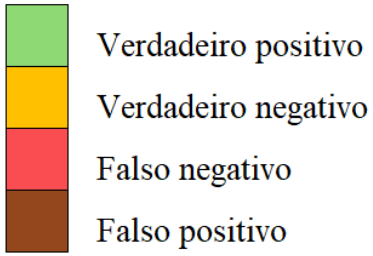
 # Primeira etapa, preparação preliminar.
Importar bibliotecas.
Baixar chave do Kaggle a partir de endereço web.
Baixar conjunto de dados a partir da chave do Kaggle e do nome do conjunto.
Extrair arquivos de áudio do conjunto de dados em uma etapa.

 # Segunda etapa, criação dos *dataframes*.
Para cada algoritmo de vetorização:
    Para cada arquivo de áudio:
        Calcular o vetor equivalente ao arquivo usando o algoritmo de vetorização.
        Inserir o arquivo em um *dataframe*.
    Salvar o *dataframe*.

 # Terceira etapa, aplicação dos algoritmos de classificação e avaliação.
Para cada algoritmo de classificação:
    Para cada *dataframe*:
        Construir um modelo de classificação.
        Separar o *dataframe* em 30% para teste e 70% para treinamento.
        Treinar o modelo de classificação.
        Testar o modelo utilizando validação cruzada com cinco repartições.
        Calcular precisão.
        Calcular acurácia.
        Calcular revocação.

Matriz de confusão

	A	B	C	D
A	120	7	9	4
B	15	116	3	6
C	12	13	115	0
D	2	96	4	38



Fonte: elaborado pelo autor

Acurácia

$$\text{Acc} = \frac{TP}{TP + TN + FP + FN}$$

Precisão

$$\text{Pre} = \frac{TP}{TP + FP}$$

Revocação

$$\text{Rev} = \frac{TP}{TP + FN}$$

Resultados

- Bom desempenho utilizando MFCC e STFT com qualquer algoritmo de classificação.
- Desempenho razoável utilizando croma com alguns algoritmos de classificação.
- Excelente desempenho utilizando STFT com *perceptron* multicamadas.
- Altura e energia inadequados para classificação.

Modelo	MFCC	Altura	Croma	Energia	STFT
Análise de discriminante Quadrático	0,91	0,24	0,80	0,18	0,82
Máquina de vetores de suporte	0,90	0,42	0,80	0,30	0,80
k-vizinhos mais próximos	0,80	0,55	0,79	0,31	0,91
Perceptron multicamadas	0,92	0,58	0,84	0,37	0,98
Bayes ingênuo	0,84	0,27	0,37	0,27	0,76
Floresta aleatória					

Fonte: elaborado pelo autor

Trabalhos futuros

- Suprir limitações deste estudo.
- Verificar o desempenho dos modelos em conjuntos de dados mais refinados e voltados para situações práticas.
- Verificar oportunidades de paralelização considerando processamento vetorial com grande volume de dados.
- Produção de um modelo para ajuste de parâmetros em cada algoritmo.
- Integração do estudo com as demais partes do sistema.

Fim.

Muito obrigado pela presença!